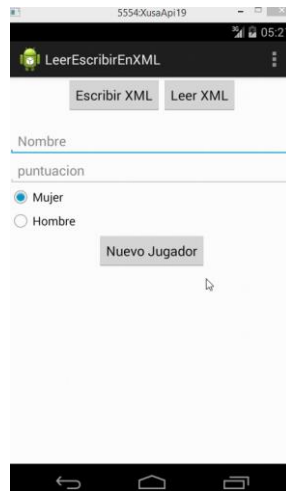


EJERCICIO PROPUESTO: LEER Y ESCRIBIR XML

Crea una app que te permita leer y escribir los datos de un jugador en un fichero .xml. El aspecto inicial de la aplicación será el siguiente:



Tendremos la clase Datos con los métodos necesarios para gestionar los datos del jugador.

```
public class Datos {  
  
    String nombre;  
    int puntos;  
    boolean mujer;  
  
    public Datos(String nombre, int puntos, boolean mujer) {  
        this.nombre = nombre;  
        this.puntos = puntos;  
        this.mujer = mujer;  
    }  
  
    public String getNombre() { return nombre; }  
  
    public void setNombre(String nombre) { this.nombre = nombre; }  
  
    public int getPuntos() { return puntos; }  
  
    public void setPuntos(int puntos) { this.puntos = puntos; }  
  
    public boolean isMujer() { return mujer; }  
  
    public void setMujer(boolean mujer) { this.mujer = mujer; }  
}
```

El Layout de la aplicación lo diseñaremos parecido al de la imagen superior. Uno de los elementos más destacables es que dentro del LinearLayout principal tendremos un FrameLayout con dos elementos: un TextView invisible que haremos visible solamente cuando pulsemos el botón para la lectura del XML, y un LinearLayout de inserción de datos funcionará de forma contraria, visible en la escritura e invisible en la lectura.

```

<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:weightSum="1">
    <LinearLayout
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="66dp"
        android:gravity="center_horizontal"
        android:id="@+id/kkk">

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Escribir XML"
            android:id="@+id/escribir" />

        <Button
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Leer XML"
            android:id="@+id/leer" />

    </LinearLayout>

<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="438dp"
    android:id="@+id/kk">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:inputType="textMultiLine"
        android:ems="10"
        android:id="@+id/contenedorSalida"
        android:layout_weight="0.94"
        android:visibility="invisible" />

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:id="@+id/contenedorDatos">

        <EditText
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:inputType="textPersonName"
            android:ems="10"
            android:id="@+id/nombre"
            android:layout_gravity="center_horizontal"
            android:hint="Nombre" />

```

```

<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="number"
    android:ems="10"
    android:id="@+id/puntuacion"
    android:hint="puntuacion" />
<RadioGroup
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Mujer"
        android:id="@+id/radioButtonMujer"
        android:checked="false" />

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hombre"
        android:id="@+id/radioButtonHombre"
        android:checked="false" />
</RadioGroup>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Nuevo Jugador"
    android:id="@+id/buttonnuvoJugador"
    android:layout_gravity="center_horizontal" />
</LinearLayout>
</FrameLayout>
</LinearLayout>

```

La información se irá guardando en un arrayList cada vez que se pulse el botón Nuevo Jugador, y pasará a serializarse a un .xml al pulsar Escribir XML. El método EscribirXml podría tener el siguiente aspecto.

```

void EscribirXml(ArrayList<Datos> datos)
{
    FileOutputStream fout=null;
    XmlSerializer serializer= Xml.newSerializer();
    try{
        fout=openFileOutput("datos.xml",MODE_PRIVATE);
        try {
            serializer.setOutput(fout,"UTF-8");
            serializer.startDocument(null,true);
            serializer.setFeature("http://xmlpull.org/v1/doc/features.html#indent-output",true);
            serializer.startTag(null,"jugadores");
            for(Datos x: datos) EscribirTag(serializer,x);
            serializer.endTag(null,"jugadores");
            serializer.endDocument();
            serializer.flush();
            fout.close();
        } catch (IOException e) { e.printStackTrace(); }
    }catch (FileNotFoundException e){e.printStackTrace(); }
}

```

```

    XmlSerializer EscribirTag(XmlSerializer serializer, Datos datos) throws IOException {
        serializer.startTag(null, "jugador");
        serializer.startTag(null, "nombre");
        serializer.attribute(null, "sexo", datos.isMujer()?"Mujer":"Hombre");
        serializer.text(datos.getNombre());
        serializer.endTag(null, "nombre");
        serializer.startTag(null, "puntos");
        serializer.text(Integer.toString(datos.getPuntos()));
        serializer.endTag(null, "puntos");
        serializer.endTag(null, "jugador");
        return serializer;
    }
}

```

El método LeerXml podría ser de la siguiente manera:

```

void LeerXml()
{
    XmlPullParser parser=Xml.newPullParser();
    FileInputStream fin=null;
    try {
        fin=openFileInput("datos.xml");
        parser.setInput(fin, null);
        int evento=parser.getEventType();
        while (evento!=XmlPullParser.END_DOCUMENT)
        {
            switch (evento){
                case XmlPullParser.START_DOCUMENT:
                    contenedorSalida.setText("");
                    break;
                case XmlPullParser.START_TAG:
                    if(parser.getName().equals("nombre"))
                    {
                        String sexo=parser.getAttributeValue(null, "sexo");
                        contenedorSalida.setText(contenedorSalida.getText()+"\nNOMBRE: "
                                                + parser.nextText()+"\nSEXO: "+sexo);
                    }
                    else if(parser.getName().equals("puntos"))
                    {
                        contenedorSalida.setText(contenedorSalida.getText()+"\n"+"PUNTOS:");
                        contenedorSalida.setText(contenedorSalida.getText()
                                                +parser.nextText()+"\n\n");
                    }
            }
            evento=parser.next();
        }
        fin.close();
    } catch (XmlPullParserException e) { e.printStackTrace();}
    catch (FileNotFoundException e) { e.printStackTrace(); }
    catch (IOException e) { e.printStackTrace(); }
}

```

En cuanto a las acciones de los botones podemos gestionarlas en el método Onclick, añadiendo a la lista un jugador cada vez que pulsemos Nuevo y guardando en el Xml todos los jugadores de la lista una vez pulsemos EscribirXml. Al pulsar LeerXml se ocultará el LinearLayout de recoger los datos y se mostrará el TextView que mostrará todos los jugadores que se hayan leído de Xml.

```

public class MyActivity extends Activity implements View.OnClickListener{
    Button leer, escribir, buttonnuevoJugador;
    EditText nombre, puntos;
    TextView contenedorSalida;
    RadioButton mujer;
    LinearLayout contenedorDatos;
    ArrayList<Datos> jugadores;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my);
        jugadores=new ArrayList<Datos>();
        leer=(Button) findViewById(R.id.leer);
        escribir=(Button) findViewById(R.id.escribir);
        buttonnuevoJugador= (Button) findViewById(R.id.buttonnuvoJugador);
        contenedorSalida=(TextView) findViewById(R.id.contenedorSalida);
        contenedorDatos=(LinearLayout) findViewById(R.id.contenedorDatos);

        leer.setOnClickListener(this);
        escribir.setOnClickListener(this);
        buttonnuevoJugador.setOnClickListener(this);
    }

    @Override
    public void onClick(View view) {
        if(view.getId()==R.id.buttonnuvoJugador)
        {
            nombre=(EditText) findViewById(R.id.nombre);
            puntos=(EditText) findViewById(R.id.puntuacion);
            mujer=(RadioButton) findViewById(R.id.radioButtonMujer);

            if(nombre.getText().toString().length()>0 && puntos.getText().toString().length()>0)
            {
                jugadores.add(new Datos(nombre.getText().toString(),
                    Integer.parseInt(puntos.getText().toString()),mujer.isChecked()));
                nombre.setText("");puntos.setText("");mujer.setChecked(false);
            }
            else Toast.makeText(this,"Debe rellenar los datos",Toast.LENGTH_LONG).show();
        }
        else if(view.getId()==R.id.escribir)
        {
            contenedorSalida.setVisibility(View.INVISIBLE);
            contenedorDatos.setVisibility(View.VISIBLE);
            nombre=(EditText) findViewById(R.id.nombre);
            puntos=(EditText) findViewById(R.id.puntuacion);
            mujer=(RadioButton) findViewById(R.id.radioButtonMujer);
            if(jugadores.size()>0)EscribirXml(jugadores);
        }
        else if(view.getId()==R.id.leer)
        {
            contenedorDatos.setVisibility(View.INVISIBLE);
            contenedorSalida.setVisibility(View.VISIBLE);
            LeerXml();
        }
    }
}

```